

a 7
additional compiling of the tagged section of byte-code by the digital information appliance,

wherein the byte-code includes at least one dynamic base object, the at least one dynamic base object comprising an interface dynamic base object and an implementation dynamic base object that communicate with each other over a message bus.

7. (Amended) A digital information appliance suitable for dynamic coupling, comprising:

a processor for implementing a program of instructions; and

a memory for storing the program of instructions, the program of instructions suitable for configuring the digital information appliance to load byte-code, said byte-code suitable for including a tagged section;

identify the tagged section of the byte-code; and

compile the tagged section of byte-code;

a 8
wherein the tagged section is compiled when the byte-code is loaded so as to enable the digital information appliance to utilize the tagged section of byte-code without additional compiling of the tagged section of byte-code by the digital information appliance,

wherein the byte-code includes at least one dynamic base object, the at least one dynamic base object comprising an interface dynamic base object and an implementation dynamic base object that communicate with each other over a message bus.

13. (Amended) A system for providing an execution environment that is suitable for dynamic compiling, comprising:

a memory device suitable for storing computer readable information;

a 9
a loader coupled to the memory device, the loader suitable for loading byte-code to the memory, said byte-code suitable for including a tagged section, the loader being capable of interpreting, just in time compiling, and pre-compiling;

an identifier coupled to the loader, the identifier suitable for identifying the tagged section of the byte-code;

a compiler coupled to the identifier;

af

wherein the identified tagged section is compiled by the compiler when the byte-code is loaded so as to enable the tagged section of byte-code to be utilized without additional compiling of the tagged section of byte-code.

18. (Amended) A method for providing an execution environment in an information appliance network, comprising:

- as*
- a) encoding an application source code in a processor independent byte-code;
 - b) tagging at least some portion of said processor independent byte-code; and
 - c) compiling at least some portion of said tagged processor independent byte-code,

wherein the independent byte-code includes at least one dynamic base object, the at least one dynamic base object comprising an interface dynamic base object and an implementation dynamic base object that communicate with each other over a message bus.

Please add new Claims 21-44 as follows:

21. (New) The method of Claim 1, wherein the message bus provides interprocessor communications within a system.

22. (New) The method of Claim 1, wherein the message bus provides communications over the network.

23. (New) The method of Claim 1, wherein the byte-code is processor independent.

al

24. (New) The method of Claim 1, wherein the dynamic base object is programmed through a scripting language with run-time object invocation.

25. (New) The method of Claim 1, wherein the interface dynamic base object and the implementation dynamic base object communicate bi-directionally.

26. (New) The method of Claim 7, wherein the message bus provides interprocessor communications within a system.
27. (New) The method of Claim 7, wherein the message bus provides communications over the network.
28. (New) The method of Claim 7, wherein the byte-code is processor independent.
29. (New) The method of Claim 7, wherein the dynamic base object is programmed through a scripting language with run-time object invocation.
30. (New) The method of Claim 7, wherein the interface dynamic base object and the implementation dynamic base object communicate bi-directionally.
- a 4*
31. (New) The system of Claim 13, wherein the byte-code includes at least one dynamic base object, the at least one dynamic base object comprising an interface dynamic base object and an implementation dynamic base object that communicate with each other over a message bus.
32. (New) The system of Claim 31, wherein the message bus provides interprocessor communications within a system.
33. (New) The system of Claim 31, wherein the message bus provides communications over the network.
34. (New) The system of Claim 31, wherein the interface dynamic base object and the implementation dynamic base object communicate bi-directionally.
35. (New) The system of Claim 13, wherein the byte-code is processor independent.

36. (New) The system of Claim 31, wherein the dynamic base object is programmed through a scripting language with run-time object invocation.
37. (New) The method of Claim 18, wherein the message bus provides interprocessor communications within a system.
38. (New) The method of Claim 18, wherein the message bus provides communications over the network.
39. (New) The method of Claim 18, wherein the independent byte-code is processor independent.
40. (New) The method of Claim 18, wherein the dynamic base object is programmed through a scripting language with run-time object invocation.
41. (New) The method of Claim 18, wherein the interface dynamic base object and the implementation dynamic base object communicate bi-directionally.
42. (New) The method of Claim 1, wherein the interface dynamic base object includes multiple lines of code.
43. (New) The method of Claim 1, wherein the digital information appliance is a thin network appliance.
44. (New) The method of Claim 1, wherein the at least one dynamic base object is fully thread safe.

REMARKS

Claims 1-44 are pending in this application.